Int. J. Nonlinear Anal. Appl. 15 (2024) 10, 359–376
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.31196.4588

Recommender systems using cloud-based computer networks to predict service quality

Mehran Aghaei[a], Sepideh Adabi[a,*], Parvaneh Asghari[b], Hamid Haj Seyyed Javadi[c]

[a]*Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran*

[b]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

[c]*Department of Mathematics and Computer Science, Shahed University, Tehran, Iran*

(Communicated by Seyed Hossein Siadati)

Abstract

In recommender systems, the user items are offered tailored to users' requirements. Because there are multiple cloud services, recommending a suitable service for users' requirements is of paramount importance. Cloud recommender systems are qualified depending on the extent to which they accurately predict service quality values. Because no service was chosen by the user beforehand, and no record of the user's selections is available, it became challenging to recommend it to users. To promote the recommender system quality, to accurately predict service quality values by offering various procedures, including collaborative filtering, matrix factorization, and clustering. This review article first mentions the general problem and states the need for research, followed by examining and expressing the kinds of recommender systems along with their problems and challenges. In the present review, various approaches, platforms, and solutions are reviewed to articulate the pros and cons of individual approaches, simulation models, and evaluation metrics employed in the reviewed techniques. The measured values in various approaches of the papers are compared with one another in several diagrams. This review paper reviews and introduces the entire datasets applied in the studies.

Keywords: QoS prediction, Recommendation system, collaborative filtering, Matrix factorization, web services in the cloud
2020 MSC: 68Vxx

1 Introduction

Cloud networks are used explicitly by everyone. Cloud service has wide-ranging uses in distributed applications and inhomogeneous environments owing to its benefits, including inexpensiveness, high reliability, and scalability. The multitude of cloud services released by service providers has encouraged users to look for superior services. In cloud-based networks, therefore, it is of paramount importance to offer a suitable service to users [13, 28, 32, 16]. Many companies and organizations have provided various cloud services to clients [33, 17, 5]. Cloud computing systems have emerged as a popular computing paradigm for hosting large computing systems and delivering different sets of services for accessing the resources virtually through the Internet, and the users pay per demand [29, 34].

*Corresponding author
Email addresses: mehran.aghaei1@gmail.com (Mehran Aghaei), adabi.sepideh@gmail.com (Sepideh Adabi), par.asghari2017@gmail.com (Parvaneh Asghari), h.s.javadi@shahed.ac.ir (Hamid Haj Seyyed Javadi)

Received: May 2023 *Accepted:* August 2023

In service-dependent environments, recommender systems mainly aim at the automatic prediction of service quality values (e.g., response time, throughput, accessibility, and reliability) of web services that users do not call for, followed by recommending the best-tailored web services to users according to the projected service quality values. In very simple terms, recommendations listed from ranked services are provided in a custom-made recommender system. Appropriate items are predicted by the system according to users' histories to accomplish ranking. However, the presence of diverse problems, including data fragmentation, implies that users have not commanded the service of interest in their histories, thus no rating or points exist for this service. Practically, using the whole service and then obtaining the service quality value is not feasible for a user. Another problem is the cold start in which data record is absent initially. The service quality is predicted using the history of users' records, and the similarity between users, services, and recent tasks is employed in the absence of history. This allows for a more accurate prediction of service quality values. A basic method for the problem-solving of recommender systems is the use of collaborative filtering for predicting missing service quality values utilizing the history of service quality records of the same users. As another prediction technique, clustering algorithms cluster users, services, and tasks and find similar items between them [36].

This review paper provides comprehensive answers to the following Analytical Questions(AQs) regarding the goals of this study:

AQ1: What are the main Subjects of recommender systems for predicting QoS?

AQ2: What are the parameters that most articles refer to and measure for comparing their solution?

AQ3: which datasets commonly use articles?

AQ4: What are the open issues and challenges in recommender systems?

Given that there are a multitude of web services, finding users' private requirements is the major question to recommend the service to users. The web service proposition according to the service quality prediction for solving the posited problem received high appreciation. Cloud service recommenders are mainly involved in enlisting new cloud services to fulfill users' requirements, offer more accurate recommendations altogether, and identify users' tastes precisely. Accordingly, finding services that could solve users' functional and non- functional requirements becomes arduous and time-taking. Service quality value prediction falls into the critical topics due to the unknown precise users' tastes and preferences and the dependence of the recommender precision upon predicting service quality values, as well as, because of the dispersal of service quality data problem, the cold start problem, and the scaling problem of available approaches. Service quality values are better predicted by recommender systems owing to their greater efficiency and precision. In recommender systems, the user and items are the two main elements [26]. A user, known as a customer as well, is the recommender system user expressing their ideas regarding the items and receiving recommendations from the system. In such systems, an item is a word referring to what is offered by the system to users. The focus of a recommender system is generally on a particular kind of item. Consequently, the customization of the design, the graphical user interface, and the underlain recommender method was employed to produce recommendations aimed at providing helpful and effectual proposals for such items. In very simple terms, recommendations as a list of items that are ranked are provided by a custom-made recommender system. The ranking is performed by predicting appropriate items by the system according to the history of users. To this aim, the history of users is collected by the system either directly via the ranking they have assigned to the items or via the user's operation inference. The mid-1990s was the emerging era of recommender systems as a study field independently, and recent years have witnessed increasing attention to recommender systems [15].

The main efforts of this research are highlighted as follows:

- Introducing the types of recommender systems and important concepts in recommender systems
- Reviewing and analyzing different methods of articles
- Summarizing and concluding and identifying open issues

2 Important Recommender system concepts

Before looking at the types of recommender systems, the inputs of the recommender systems are reviewed below.

2.1 Inputs of recommender systems

The input of a recommender system depends on the type of applied algorithm. The input of the recommender system is described as follows:

2.1.1 Rating

Users rate the items. These scores must be in a numerical range defined in the system. For example, from 1 to 10, where 1 represents the worst and 10 represents the best. In addition, rates can be collected indirectly

2.1.2 Demographic

This type of data includes information related to users such as their age, gender, occupation, and education. It is difficult to collect such data implicitly, and it is usually collected explicitly from users.

2.1.3 Content

This data is not collected directly, but processed by analyzing the data related to the items. Features extracted using this method are used as input to a filtering algorithm to extract user profiles.

2.2 Type of recommender systems

Recommender systems need to have the ability to identify items that provide benefits to users and predict whether an item or service is worthy of recommendation to them. To this end, the system needs to have the ability to make a comparison of the practicality of items or leastwise compare the practicality of some items, followed by deciding on recommending some items according to this comparability. For the mentioned reasons, various approaches are used by recommender systems for predicting the opinions of users about items. There are six groups of such systems (Fig 1).

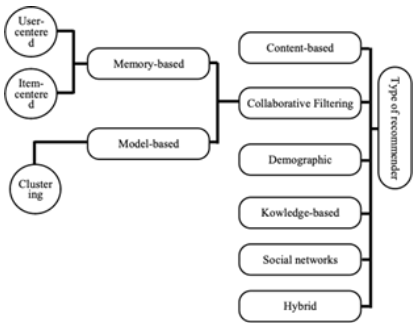

Figure 1: Type of recommender system

2.2.1 Content-based

In this approach, items are suggested by the system to users that are the same as those chosen by them in advance. The characteristics assigned to items are used to calculate the similarity between them. The dependence of content-based filtering is on the present users' information and tastes, such that suggestions are provided by considering their previous selections and experiences. These approaches generally need the analysis of user-related information and contents and the items in the system to the ability to ascertain the similarity level between the user and the items of the system, as well as between the items. The results recommended to users contain items with more similarity to those chosen previously by the user. In these systems, the cornerstone is to select a measure of similarity. In content-based filtering, the content of the considered item is also used to make recommendations, thus, making it possible to use specific content including film, photo, and sound. As such, the items with the same content and features as those procured, chosen, or ranked by users are proposed to them [9].

2.2.2 Based on Collaborative Filter (CF)

In a very simple event, items are suggested by this approach that users are interested in the same as active users. The users' opinions formerly assigned to the items are used to calculate the similarity between them, which is examined thoroughly in this paper.

2.2.3 Demographics

In such systems, items are proposed according to the information obtainable in the profiles of users. For instance, an application of this technique is upon sending users to special websites depending on their languages and countries.

2.2.4 Knowledge-based

In knowledge-based recommender systems, items are suggested according to the extent of knowledge regarding meeting users' requirements and preferences by the features of a specific item, as well as the utility of the item to users. In this method, the extent to which the presented proposals correspond to users' requirements is estimated by the similarity function.

2.2.5 Recommender social networks

Along with the development of social networks, the information existing in these networks was utilized in recommender systems by some investigators. Such systems propose items to users as preferred by their friends. It is documented that people confide their friends' commendations more than those of the same but anonymous people. It is possible to collect this information in explicit or implicit ways. The results for the use of this information determine that this work has led to an improvement in the suggested results and a reduction in the data dispersal problem. To utilize social networks in recommender systems, investigations, and research projects are categorized into two groups [9, 44]. In a category, the information existing in these networks is used to make the existing systems more efficient, and their findings demonstrate that this information positively affects recommender systems. Instead, a recommendation system according to collaborative filtering was established by another group of scientists. The aim of this second group is not to integrate social networks into other recommender systems. Instead, they intend to utilize the potentialities in these networks to produce an independent system. Trust and popularity have received more attention than other information existing in social networks. Trust measures a user's trustiness level among the rest of the users, which can effectively influence proposals. For instance, users' greater trust levels amplify their opinions given to items in terms of importance and weight levels than others [22]. Users' trust is determined by two methods, the first of which is via information gained overtly from users. In the second procedure, information and associations between users existing in social networks are used implicitly. Up to now, the solutions recommended in studies include trust propagation mechanisms, trust networks, similarity criteria based on individual characteristics, and distrust analysis. The popularity measure is employed for items and it is determinable for items according to the number of opinions given by users to an item (explicitly) or by examining users' working procedure with items (implicitly) [23].

2.2.6 Hybrid recommender systems

The aforementioned methods are combined in these kinds of recommender systems. A combined system mixes methods A and B aiming at taking advantage of method A for eliminating the drawbacks of method B. Using a hybrid approach offers better proposals and recommender systems are used for more complicated applications such as commercial applications. In hybrid recommender systems, multiple algorithms are combined for achieving better performance and taking advantage of and benefits of one technique to surmount the deficiencies and drawbacks of other approaches. In combined user models, collaborative filtering is usually mixed with content-based or social network-based filtering [2].

2.2.7 Collaborative filtering details

The collaborative filtering method, known as the basis of work in numerous other solutions, accounts for the major and commonly applied filtering techniques in recommender systems. This algorithm works similarly to what is acted in our daily decision-making. For instance, we purchase a product that is more popular with other people (e.g., it has received more opinions in the e-commerce world). Thus, it can be argued that others' experiences instead of that of the individual are considered in the collaborative filtering method. This technique requires users to be initially permitted to take part in the system and rank the different items in the system. Proposals are presented according to the users' similarity to the target user or items ranked by the target user. The idea behind collaborative filtering is to choose k of the greatest close neighbors for the user, followed by locally predicting the uncertain QoS values [20]. Since this method is simple and usable, it is the service quality prediction technique with the utmost popularity in service presentation. Inherently, the problems of this technique are data latency and cold start, particularly in dynamic service-oriented environments. Alternatively, CF falls into the QoS prediction approaches with the highest reputation in RSs, being utilized in offering precise services to active users [25]. Collaborative filtering consists of the memory-based and model-based approaches, as described below.

2.2.7.1 Memory-based algorithm

In the memory-based model, it is assumed that users or services sharing a similar QoS history probably share the same QoS values in the future. Every user is assigned to a group of users with the same interests. This technique includes first loading the data accessible in the memory, followed by making the predictions based on the existing data, and the users are identified the same as the target user, known as neighbors. In this technique, QoS history records are necessary and all the QoS matrix is used for this. The efficacy of this technique depends largely on measuring the similarity, for instance, Pearson's correlation coefficient and Cosine similarity. In the memory-based cooperative filtering method, the most widely used algorithm is the KNN algorithm, which contains two approaches [8]:

2.2.7.2 User-centered approach

The user-centered method initially calculates the similarity of different users according to a similarity measure (Pearson correlation coefficient, cosine criterion, etc.). Then, new votes are predicted for individual users according to the values of the votes for some of the most similar people to V (in terms of the target user's neighbors) based on the square matrix of users' similarity. Finally, the target user is proposed with a list of N items with the most predicted votes.

2.2.7.3 Item-oriented approach

Contrary to the former technique, the item-oriented approach initially calculates the similarity between the system items, followed by calculating new votes for individual items according to the votes given to its adjacent items based on the square matrix of similarity. Lastly, individual users are recommended the items with the utmost votes. In the memory-based technique, recommendations are made more accurately, but the rise of users and items rapidly elevates the calculation time. Several cases make it difficult to answer in actual time. The user-oriented technique suffers from two fundamental problems: low scalability and vulnerability to data fragmentation and cold start in the database. In the case of scalability, adding a new user to the system requires recalculating the similarity measures and prediction values, thereby increasing both the number of users and the size of the system, bringing lots of computational overheads to the system and causing problems. In the data dispersion and cold start problem, the calculated similarity may be unreliable because there are inadequate users' opinions about the items [43].

2.2.7.4 Model-based method

Also called the factorized matrix (MF), this model produces a learning model using data obtained from the user-service matrix and learning the factors hidden in the matrix for predictions in the future. In this approach, the default is that services or users are independent. Consequently, it fails in finding similarities between various users or differing services. This approach is usually high-cost and time-taking. It creates a complex model according to the training data set and machine learning approaches and can predict the service values by merging the data history of the same users [23].

In this group, the employed procedures include neural networks, Bayesian categories, clustering, linear regression, fuzzy systems, Markov chain-based models, etc. One of the best-known model-based approaches is the factorial matrix, where the user-item score matrix is grouped into small user-feature and item-feature matrices. Next, the prediction is made by internally multiplying the user feature vector and the item feature vector. Model-based methods show good performance with wide-ranging datasets [12].

2.2.7.5 Methods based on clustering to solve model-based problems

An alternate method to model-based approaches is clustering algorithms, which contrary to the former approaches working with matrix decomposition, diminish the search space by clustering and grouping users or services. As an advanced classification technique, clustering algorithms mainly aim at finding rapidly the group of users or services with utmost similarity. It is worthy of note that clustering algorithms cannot acquire a personalized set of clustered users based on available similarity criteria.

According to clustering, two general groups of recommenders are available [3]:

1- Cummins

2- Comediodes

2.2.8 Challenges of recommender systems

In the context of recommender systems, multiple challenges influence the efficacy of algorithms and techniques utilized in this field. The following are the major challenges [15]:

2.2.8.1 Dispersion of data

Because the sites contain numerous items, users cannot rank the whole items. The user-item matrix will be empty; hence, it is necessary to consider data confidentiality in the design of efficacious recommender systems.

2.2.8.2 Scalability

Because many calculations needed for implementing recommender systems increase with increasing users and items, high scalability is needed for the proposed algorithms.

2.2.8.3 Cold start

Inadequate information that precludes producing validated suggestions leads to the cold start problem. Upon the entry of new users into the site, they may be presented with an offer because of insufficient information. Hybrid recommender systems fall into the approaches proposed for solving this problem. Newly added items to the system will not be presented to the user as they are still unrated.

2.3 Similarity calculation criteria in collaborative filtering

In recommender systems, the work according to collaborative filtering is to ascertain the similarity level between users and items to be able to obtain the neighborhood accordingly. The precision and performance of recommender systems are improved by selecting a proper similarity criterion. Traditionally, the similarity was ascertained according to scoring. The similarity between users is estimated according to the information of the user-item score matrix (consisting of the similarities of users to items). Therefore, voting two users for a single item leads to the system's conclusion that a similarity exists between the two users. This issue can be determined differently in the case of items, such that voting a user for two items results in the system's assumption that a similarity is present between the two items. In addition to this measure, the similarity between users and items is determined using other cases. A few of these measures include Pearson's correlation coefficient, cosine criterion, modified cosine criterion, etc. [4].

In collaborative filtering algorithms, an essential factor is to appropriately choose a similarity function due to its effect on the accuracy and performance of the suggestion. In memory-based collaborative filtering, the similarity is calculated based on common similarity measures by the nearest neighbor algorithm. These measures only need information sources offered by users or items. In memory-based collaborative filtering recommender systems, the employed similarity criteria are summarized in the following table [12]. The table represents that two score vectors are correlated linearly. The value of this measure ranges from -1 to 1, with the value of A, zero, and -1 meaning complete positive correlation, no correlation, and complete negative correlation, respectively. This measure assumes the angle between two score vectors, with a smaller angle indicating higher similarity. The creation of this criterion aimed at compensating for the deficiencies of the cosine similarity criterion. Items may be rated by users on differing scales; for instance, an item may not be very interesting to users, but they tend to rate it at a high level and contrariwise. In the modified version, Pearson's correlation coefficient is considered to allow having only similar sign pairs of scores. To improve correlation, for instance, both points must be positive or negative. The idea of trust and credit is used in this version. If common items are assigned more points by two users, they will show more valid similarities. The low similarity problem of common items between users may be declined using the Pearson correlation coefficient based on the sigmoid function [35]. In this function, only the number of points is considered that is shared between two users, who are more similar if they share more points [31].

- Mean squared difference criterion Only the number of points assigned to items (the number of points allocated to items by users) is considered in this criterion.
- Combined Jacquard criterion and mean square difference

This criterion combines Jacquard features and mean squared difference. The development of this hybrid measure only aims at solving Jaccard deficiencies and mean square differences to some extent. It comprises three factors: proximity, influence, and popularity. Besides calculating the difference between the absolute values of the scores, the proximity factor also examines the acceptability of these scores, penalizing unreliable scores. The influence factor indicates the extent of users' interest in an item. The popularity factor is used to represent the reason for the same rankings by two users. More information is obtained about the users' preferences when the mean score difference between two users is far from that of the users' total score. The absolute value of points (number of points), the absolute value of points (number of points), and the points that users have given to common items are generally considered by this measure. Additionally, the local content information of the scores is used by this measure [19].

In an analysis of the drawbacks of the current similarity criteria, Elyo et al. concluded that similarity criteria generally suffer from many difficulties, which are partially listed below [6].

- Number of common items: There is a low number of common items ranked between two users.
- Cold Start: A small number of items ranked by a new user.
- Smooth scores: A small number of items given a similar score by two users.
- Scattering of the user-item matrix: A high number of items not scored by users.
- In a few papers, other drawbacks of similarity measures are considered as follows:
- Failure to correctly calculate similarity, when two users give similar points to common items.
- Inability in the correct calculation of similarity when common items are assigned differing points by two users.
- Reduced accuracy as it ignores the proportion of shared points.
- It will be difficult to discriminate various users by overlooking the absolute value of points (number of points).
- The user-item matrix's cold start problems and sparsity cause very low similarity accuracy.
- The focus on the local content of the scores and disregard of the overall preferences of the user's behavior can mislead similarity.

The focus of Elu et al. was on the performance and efficiency of proposals in memory-based collaborative filtering algorithms. By introducing a novel heuristic similarity criterion, named NHSM, the design of which is based on the PIP criterion, the authors aimed at solving the deficiencies and problems of the current similarity criteria. The original PIP similarity is abnormal and is calculated with complication. Thus, the NHSM similarity model proposal aimed at overcoming such problems. The percentage of shared points between two users has improved by the novel similarity criterion. They investigated various users with differing scoring preferences using the average and variance of scores to represent the users' score preferences. The conclusion of their experiments was that the novel similarity criterion could achieve the greatest performance in memory-based collaborative filtering by overcoming the problems and drawbacks of present similarity criteria. Proximity, importance, and uniqueness are the three factors of the NHSM similarity measure. The proximity factor consideration is the distance between two points. As the second factor, the importance indicates more importance of two scores if they are more distant from the mid score. For instance, if the points of two items are (4, 4) or (2, 2) by two users, their importance is higher than two users who have assigned two items' points of (5, 3) or (2, 4). Uniqueness is the third factor showing the difference of two points from other points. Additionally, NHSM merged modified jacquard similarity with user-score preference in design. As mentioned above, the original PIP similarity formula suffers from high complexity and non-normality. The novel similarity model uses a nonlinear function, named the sigmoid function, to prevent bad similarity and utilize good similarity. The improved version of the PIP measure is termed the PSS measure. The following shows the user's PSS similarity calculation. In the PSS measure, there are three similarity factors: proximity, importance, and uniqueness [21].

3 Review and analysis

Data fragmentation and cold starts are the drawbacks of numerous recommenders. In basic clustering approaches, only obtaining the similarity between users or services is possible, and they fail to discriminate the similarity of tasks from various users because dissimilar tasks may be created by the same user. Thus, the result of the prediction will differ by disregarding this issue. Cloud computing-based manufacturing (CMfg) mainly aims at intelligently virtualizing distributed resources and their capabilities, followed by their encapsulation as a cloud service for facilitating the access of users to current solutions. In the CMfg platform, the same functionality is shared in numerous cloud services. This problem leads to the high importance of the cloud recommender system.

Clustering and trust methods were merged by Liu et al. [40]. The algorithm mainly assumes that highly similar users plausibly have the same tastes and ratings (QoS value). The similarity of users is obtained by first collecting explicitly written information and implicitly obtained information from the rating. They were the first to mention the similarity of tasks. They computed the similarity between the tasks by the Comediode clustering algorithm. It is possible to obtain the QoS values from undependable users. Actually, a technique should be provided to make users' data trustable. In this article, local and global trust values were combined to design a trust-based collaborative filtering technique, which can reconstruct the trusted network of clustered users. Finally, the clustering algorithm and the trust technique are combined to produce a personalized service quality prediction and achieve a trustworthy service commendation for active users in CMfg. In the reviewed article, a new clustering-based technique is employed that utilizes task similarity for identifying similar users [21].

In Liu et al. [27], a technique is introduced that principally uses the improved cosine similarity technique to obtain the similarity of users. This technique is set by the cloud service popularity. Subsequently, multiple similar users are chosen as the user's neighbors. Lastly, the prediction and recommendation of services are based on the user's neighbors. The focus of most former web service recommendation approaches is on non-functional requirements and quality of service (e.g., response time, throughput, accessibility, and reliability). This introduced technique mainly considers recommending the cloud service with the utmost service quality values for users. The task mainly predicts the unpredictable quality of service values with the cooperative filtering procedure. Nevertheless, suitable and precise results are obtained when they can first fulfill the users' functional requirements followed by non-functional requirements. The users' functional requirements were met using some content-based tasks and procedures. In the content-based technique, users' personal preferences are explored by the analysis of cloud service description information.

Nonetheless, the commendation is less accurate because there are differences between the performance of the cloud service, the service description information, and the complex analysis of textual information. The collaborative filtering technique can effectively specify users' preferences by the direct discovery of users' application history.

The taste of users is explored using Pinhadi's technique consisting of three phases

1. Calculating the similarity of users
2. Selecting the neighbors of users
3. Forecasting

In this technique, the suggestion accuracy is dependent on detecting the user's neighbors. This is achieved by the user's similarity calculation technique and the user's neighbor selection approach. The similarity calculation is improved via an introduced weighting factor for cloud service popularity in cosine similarity, and a parameter is produced and measured to filter the user neighbors. The good performance of this technique is relative to other papers in the F-measure parameter was proven by evaluating the results. The improved efficiency of the presented technique is generally owing to the similarity calculation procedure and neighbor selection approach.

Cao et al. [10] used the time service quality prediction (TWQP) method, comprising two phases, for solving recommender system problems. The first and the second phases are based on the history of time slices and the current time slice, respectively. When a service in the previous time slice is called by a user, the service quality value for this user, who calls this service next time, is forecasted according to the service quality data history. If a service is not called by the user in the former time interval, the uncertain values of the service quality are predicted by the CA algorithm. In the second phase, the results of the former phase are used to predict the uncertain values of the current time slice. In this technique, users are generally selected who have requested the same services as the target user u.

In Career et al. [11], an innovative technique is proposed based on the temporal correlation coefficient and Cummins enhanced with Cocoa search, named TCCF in this article. In the clustering technique, the same users are categorized for accurate and rapid commendations in the future. They also developed an effective and personalized recommendation based on the preferences model (PTCCF) for the quality improvement of TCCF, offering more quality recommendations by user behavior analysis. In a novel service quality prediction framework, designed by Feng et al. [7], uncertain service quality values are predicted using both location and neighbor information. This framework mainly consists of the following parts:

1. CMfg Service Platform: Service quality is predicted by collecting and recording the profiles of users and services.
2. User-service matrix of service quality: Service quality values are recorded users' feedback from services. The history of M users and N services are recorded in an $M \times N$ matrix in which every row denotes the service quality value of the service j of user i.

3. Similarity calculation: The neighborhood information of users and services is acquired according to the former matrix. Pearson's correlation coefficient similarity: The basic similarity calculation tool was the use of Pearson's correlation coefficient. Obviously, the similarity calculation was also strengthened using some modifications.
   - Spatial similarity: The profile of users and services in the CMfg platform, including country, location coordinates, IP address, etc., are used to obtain this component.
   - Similarity setting: The number of the same services selected by two users is regarded for the neighborhood of two users.
   - Topology similarity: A graph is formed when a user selects a service. Services and users are nodes, and the connections between them are edges. The graph structure is the basis to find similarities between nodes.
4. Similar neighbor selection: In this paper, the quality of uncertain services is obtained by considering the most similar users and services for finding the neighborhood set using collaborative filtering for this purpose.
5. Neighbor-enhanced factorized matrix: Uncertain service quality for the service is determined by several users, thus the factorized matrix model is used for this purpose.
6. Predicting the quality of uncertain services: The calculation of uncertainty values is based on the training set derived from the improved matrix.

To help access appropriate services for active users, a cooperative service quality prediction technique, termed context-sensitive factorized matrix, was offered by Aggarwal et al. [1]. Not only users and services are included in service quality values, but this technique also takes service context information into consideration. Contextual information, including time, geographical coordinates, network conditions, device settings, and similar information, are mentioned in this method. Here is a matrix with three user, service, and context dimensions that represent service quality values, i.e. the conditions and context of the service are also clear in the service choice by users. The effectiveness of factorized matrix procedures is typically higher than user-based and item-based collaborative filtering as the former finds concealed features between the user and the service. In this paper, a noteworthy point is that service selection not only depends upon the client and server but also on the conditions and communication quality in the network. The successful application of deep learning based on hybrid methods has recently been reported in recommender systems. The use of deep learning in recommender systems can be attributed to the high capability of learning concealed structures from the interplays between the user and the service. Ekstrand et al. [14] integrated and combined interplays between user and service to propose a new technique for web service recommendation. Apparently, deep learning based on hybrid approaches is more helpful than collaborative filtering. Such approaches are capable of learning concealed interplays between the service and the user. However, they fail an accurate understanding of the implied information between the user and the service. Altogether, collaborative filtering and textual content are merged in this method. The context function in the neural network is used to determine the choice between mashups and services.

Guy et al. [37] presented a hybrid method for predicting future and uncertain values of service quality. To cluster the dataset, the manipulated Comediode algorithm and a method based on lazy learning and LB-Keogh for pruning are combined in this method, which fills uncertain QoS values with pseudo-real values. There are two offline parts and one online part in this paper. The first offline part consists of preprocessing and clustering, and the online part comprises prediction. The future can be predicted using new data. In this article, differences between service usage histories are detected using time series clustering. For training, service quality values were predicted using the manipulated lazy algorithm.

Shinde et al. [39] utilized the graph structure representative of various data types and their associations. The concealed association between users and services is determined by deep mining. They then divided the graph into stronger subgraphs with more meaningful connections by cutting meaningless edges and lowering noise. Finally, their prediction of service quality values was made by the fusion model to merge local and national information. Graph creation, graph decomposition, and adaptive composition are the three main parts of the proposed framework. In the first part, the graph is created and drawn, integrating various information sources, including latitude and longitude, functional service description derived from WSDL, and user score acquired from the service quality matrix. The graph formation is based on the similarity or weighted association of the edges. The second part places similar nodes, including user nodes and service nodes with the utmost associations, in their subgraphs. The third part serves to predict service quality values based on the local data of the subgraphs, and the global service quality values are predicted from the initial graph according to the improved probabilistic factored matrix algorithm. Next, the final prediction of local and global values is based on using a Gaussian combination. The following introduces a few related papers, nearly all of which have compared their work. These are fundamental papers in the context of recommender systems. Since the papers are somewhat old, a brief discussion has only been made on them.

In Kosmides et al. [24], a technique based on cooperative filtering, termed UPCC, was developed to identify similar users. They employed Pearson's correlation coefficient to this aim. The history of service quality records is the input data of this algorithm. In this system, the uncertain quality of service values is predicted using the QoS values of the user's neighbors. As with the aforementioned technique, Sharma et al. [38] proposed an IPCC method that utilizes the neighbors' service quality values using similar services for this task for predicting the QoS values. This means that it acquires the user's service quality values from the service similarity of the neighbors' service quality values. Some researchers investigated the method developed based on the factorized matrix model. A service quality prediction model based on service composition was reported by Middleton et al. [30]. The technique uses user information and the history of service quality records to calculate the uncertain service quality values end-to-end using these data, but this technique takes a very long time. They converted and encoded the input data into vectors in their article, followed by using the embedded layers to diminish the input dimensions and determine the connection between the input vectors. In the second step, the output of the embedded layers was trained by the factorization machine. The first-order and second-order weights were assigned to the factorization machine as base weights. The multilayer perceptron was then trained based on this, followed by determining the parameters of this model using the history data. The model used these new data to generate predicted QoS values. The collaborative filtering-based technique is divided into two approaches based on rating and ranking. The rating-based model predicts the rating of uncertain service values according to the previous experiences of similar users. In the ranking-based technique, the score of the services is predicted to ensure the prediction accuracy. All these approaches possess their own benefits. -In the first technique, the prediction is made by ignoring the final prediction accuracy based merely on the history of records and similar users, and the second technique prediction is based on the final prediction accuracy. In [18], these two approaches were merged to benefit from both. In this technique, the predicting process needs trusted users' selection to reach more accurate forecasting. In this article, a hybrid and improved similarity measure is applied that employs trusted and related users.

Deep learning was used for forecasting service quality values [42]. This approach mainly aims at combining MF matrix factorization and clustering techniques based on geographical features. This paper solves the data dispersion problem by clustering the primary service quality matrix into sub-matrices including geographic information and service providers. For training, a neural network, termed Deep Autoencoder DAE, is used in every sub-matrix. This article also utilizes SOM self-organizing map clustering since it is very advantageous, including maintaining the topology of QoS data. Grouping the trained neurons in the map is based on k-means. By adding some random data and noise to the input data, they reduced the sensitivity of the model to overfitting, which may jeopardize the final performance.

Using collaborative filtering, matrix factorization (MF), and collective learning, Sun et al. [41] detected probability distribution based on combined prediction approaches to attain correct and personalized web service recommendations. A complex and high-cost novel model is not designed and presented in this technique. To predict the quality of uncertain service values, k neighbors around the user are used in collaborative filtering. This technique is advantageous as it is simple and usable, and its drawbacks are data scattering and cold start. In the factorized matrix, on the other hand, machine learning is applied for understanding the concealed association between the user and the service by multiplying the two user and service matrices. A benefit of this technique is that it solves cold start and data dispersion problems, and some drawbacks of this approach include its inconsideration of the similarity between users and services, time-consuming, and high cost.

Feedback and service context are included in this section. Computing performance, memory, and network facilities are clustered under cloud service infrastructure. These items are regarded as the service context. One can specify the context of the service by reading the service description information. There is a specific description, named the description or service agreement level, for every service provider. The functionality of the service is described by context attributes. Thus, the proposed algorithm is employed to consider the service context and perform similarity calculations. Moreover, users are allowed to post feedback about their used services using the feedback subsystem. As with the ranking concept in the collaborative filtering method, the service quality values of unrated services in the factored matrix are predicted using users' feedback. The required user feedback for some QoS factors and processes is recorded in this article using the mapping and reduction framework to diminish data fragmentation. These problems were solved using the factored matrix principle, and the system's quality was enhanced by forecasting the service with the maximum service quality values. This subsystem determines service details, including service nature, service type, service capabilities, and usage cost, by the service crawler such that the provider site determines the service description by navigating the service. Basic details about the services are revealed by these descriptions, along with updating the service repository.

The studied methods are listed in Table 1, including writers, year of publication, article title, method name, research method, advantages and disadvantages, evaluation metrics, and dataset used in the article.

Table 1: Reviewed articles at a glance

Writers	Year	Title	Method name	Research method	Advantages and Disadvantages	Evaluation metrics
Jian Liu et al.	2019	Personalized clustering and trust-aware service quality prediction method for recommendation of cloud services in cloud production	clustering-aware and (CTP) trust-aware approach	The K-medoids clustering algorithm uses explicit textual information and implicit ranking information to determine the similarity of users. Combining local and global trust values and reconstructing the trusted network of clustered users led to the developing of a trust-aware collaborative filtering technique.	Identifying the similarity of tasks among different users Solving the problem of data fragmentation Address the cold start problem Considering the lack of computational complexity and processing power of cloud clusters	MAE, NMAE, F1
Lee et al.	2020	Detecting a probability distribution using the hybrid method of ensemble service quality prediction (DHEM)	DHEM (Detection based Hybrid ensemble Model)	Inspired by collaborative filtering, factorized matrix, and ensemble learning, this paper conducts probability distribution detection based on the combination of prediction methods to generate accurate and personalized web service recommendations.	Accurate calculation of user similarity; High prediction accuracy but costly due to the need to learn; Better performance on more sparse data; The weight parameter must be set manually.	MAE
Nagarajan et al.	2019	Cloud infrastructure service recommendations based on context-aware service quality prediction as a service	service context-aware matrix factorization (SCMF)	1- Proposing a cloud broker infrastructure that includes a user portal for collecting feedback and a recommendation process 2- Extracting service particulars and context values for evaluation 3- Calculation of service similarity and clustering according to service quality values 4- Predicting uncertain service quality values using the factored matrix based on contextual information.	Response time enhancements for sparse datasets The cost of calculations is increased due to preprocessing and attaining higher levels of accuracy. A minor decrease in the error rate in the permeability matrix decreases the error rate at the beginning but increases it at the end. Response time and throughput enhancement in sparse matrix Reduction in the extent of the dataset as a result of preprocessing based on the service context	NMAE, RMSE
Wang et al.	2019	A collaborative filtering method for cloud service recommendation through usage history exploration	CFR (Collaborative Filtering Recommendation)	Using an enhanced cosine similarity method to identify user similarity	It could not solve the cold start problem. Better performance of F-measure	F-measure
Jin et al.	2020	Time-aware dynamic QoS prediction for services	Time-aWare service Quality Prediction method (TWQP)	Solving recommendation system problems using the TWQP	The most important advantage of this method is that the service quality change over time.	MAE, RMSE
Cui et al.	2020	Personalized recommendation system in IoT scenarios based on collaborative filtering	personalized recommendation model based on preference pattern (PTCCF)	The primary objective is to provide recommendations based on the similarity of user preferences. Collaborative filtering analyzes the preferences of users by their behavior (user behavioral recordings). Then, it makes recommendations based on the likes of individuals with similar interests.	- A comparison with the number of articles has been conducted. - Contrary to most articles, data dispersion was explicitly measured. - Considering the large volume of data, data growth, and multidimensionality are among this method's advantages.	MAP, F-measure, MAE
Feng et al.	2018	Prediction of service quality of cloud construction service based on neighborhood-enhanced matrix factorization	NEMF (Neighbourhood Enhanced Matrix Factorization)	Introducing a framework using spatial and neighbor information to predict uncertain QoS values	Fixing the overestimation of the PCC method Fixing cold start and data sparsity	MAE, RMSE
Wu et al.	2018	Prediction of collaborative service quality using context-sensitive matrix-factorization	CSMF (context-sensitive matrix-factorizatio)	CSMF collaborative service quality prediction method helps active users to access appropriate services.	Solving data sparsity problem Poor performance in dense matrices Explicit and implicit information use	MAE, RMSE, NMAE

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

4 Discussion

In this section, according to the reviewed articles, three topics are explained. First, the metrics and parameters used by the articles to measure their method are described. In the second part, the datasets used in the articles are

Xiong et al.	2018	Deep hybrid collaborative filtering for web service recommendation	DHSR (Deep Hybrid Service Recommendation)	Using deep learning	Similar to the factored matrix method, it has the problem of data sparsity; Prediction accuracy is low in the face of high computational complexity.	MAP, NDCG, Precision, Recall, F1
Keshavarzi et al.	2020	Enhanced time-aware service quality prediction in cloud network: Hybrid of K-medoids and lazy learning	QoPC	A hybrid of manipulated K-medoids algorithm is used to cluster the dataset, A method based on lazy learning, and LB-Keogh for pruning.	Solving the cold start problem Slowness due to preprocessing Higher accuracy due to preprocessing and removal of noise data	MAE, NMAE
Chang et al.	2021	A graph-based quality of service prediction method for web service recommendation	GMF (Graph-based Matrix Factorization approach)	The latent relationship between users and services is obtained by deep mining in the graph and forming strongly connected subgraphs representing the significant relationship between the service and users.	It is a suitable method for the hybrid of different data sources. It has a data sparsity problem	MAE, RMSE
Zheng et al.	2010	Distributed Quality of Service Assessment for Real World Web Services	UPCC (User-based prediction algorithm using PCC (Pearson correlation coefficient))	A base model for identifying comparable users. It was accomplished using Pearson's correlation coefficient. Service quality recordings constitute the data. This system predicts the uncertain quality of service values using the neighborhood quality of service values.	The test was conducted on a small scale.	Response time, Throughput
Xiong et al.	2014	Predicting web service quality collaboratively on distributed data	item-based collaborative filtering methods (IPCC)	A base model that predicts service quality values using neighborhood service quality values and employs similar services and derives the service quality values of the user from the service similarity of the neighborhood service quality values.	Data distribution is unbalanced.	MAE, RMSE
Chen et al.	2014	Web service recommendation based on the quality of service using CF	UIPCC	A new method was created by combining IPCC and UPCC: a collaborative hybrid filtering model called UIPCC.	The test was conducted on a small scale. It has the basic problems of collaborative filtering.	MAE, RMSE
Karim et al.	2016	Using user and service data and latent factors to predict service quality for service selection and recommendation in hybrid service	-	Uncertain service quality values are calculated end-to-end utilizing user information and a history of service quality records.	It is incredibly time-consuming.	MAE,RMSE
Milad Ahmadian et al.	2023	Knowledge-aware recommendation system by improving group trustworthiness	Reliable deep ensemble reinforcement learning-based recommender system(RDERL)	This article emphasizes the recommender system's reliability and uses a deep neural network and reinforcement learning.	Ignoring factors such as complexity and overhead	Precision, Recall,F1, NDCG, DCG
Ganan Gaskaran et al.	2022	Utilizing time series prediction and deep learning techniques to evaluate service quality prediction for web service recommendation	CF-MkM-CNN-LSTM	Considering the service call time by the user and using deep learning algorithms	Higher accuracy than compared methods and reduced time cost	MAE, RMSE, Precision, Recall
Khani et al.	2023	A deep hybrid approach using side data in recommender systems	A novel deep hybrid side information-based recommender system(DHSIRS)	Determining nonlinear latent relationships between user and item through deep learning model and using a multilayer perceptron neural network to combine side information and user-item interaction matrix.	Solving cold start problem, high flexibility	MAE, RMSE, NDGC
Liu et al.	2023	Improving the performance of cold-start recommendation by fusion of attention network and meta-learning	Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning(AMeLU)	Using a meta-learning method to reduce cold start errors and using a hybrid of meta-learning method and attention mechanism to increase the ability to personalize user interest	This learning model necessitates a limited training group. This model considers an extensive range of user interests.	MAE, RMSE
Sangeeta et al.	2023	Deep knowledge graph based attribute preserving recommendation	Deep Knowledge Graph based Attribute Preserving Recommendation (DKG-APR)	It obtains intricate graph connections using deep neural networks and attention mechanism. The data is collected as a stream.	Better performance in graphs with high communication complexity	NDCG, Recall

[a]Gaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

described and at the end of this section, the methods used in the studied articles are compared in the form of graphs on some metrics.

5 Evaluation parameters

In this section, the metrics presented by various articles to evaluate and compare their work have been compiled.

The metrics that have been the most referenced in the articles are as follows.

A total of 12 parameters are introduced in this section, which is shown in Table 2.

Table 2: Most referenced parameters for evaluation used in articles

Title	Formula	Description
MAE (Mean Absolute Error)	$MAE = \frac{\sum_{i,j}(R_{ij}-\widehat{R}_{ij})}{W}$	The MAE variable displays the difference between predicted and actual values. Rij is the actual value of the quality of service that the jth user gives to the ith service. W is the number of predicted values of service quality.
NMAE (Normalized Mean Absolute Error)	$NMAE = \frac{MAE}{\sum_{i,j} Rij/W}$	NAME is a variable obtained from the previous scarecrow and used to measure prediction accuracy.
F1	$F1 = \frac{2*Pr*Re}{Pr+Re}$ $Pr = \frac{1}{\lvert U\rvert}\sum_{u\in U}(rel_{u,N}/N)$ $Re = \frac{1}{\lvert U\rvert}\sum_{u\in U}(rel_{u,N}/rel_u)$	F1: recommended quality and hybrid of (Pr) and (Re). Pr: Probability that the selected service is relevant or required. Re: Probability that related or required service will be selected. |U|: shows the number of test users. Relu: the number of services related to the user (u) Relu, N: number of related services in N number of highest recommendations
RMSE	$RMSE = \frac{1}{W}\sqrt{\sum_{i,j}\left(R_{ij}-\widehat{R}_{ij}\right)}$	RMSE is a measure of prediction accuracy.
Normalized Discounted Cumulative Gain (NDCG)	$NDCG_N = DCG_N/IDCG_N$ $DCG_N = \sum_{i=1}{}^{N} \frac{2^{rel_j}-1}{log_2(j+1)}$	$NDCG_N$ used to measure the ranking prediction accuracy. The higher the value of this parameter, the higher the accuracy. Ideal Discounted Cumulative Gain (IDCG) Discounted Cumulative Gain (DCG) relj is rating of service which located in position j.
MAPE (mean absolute percentage error)	$MAPE = \frac{\sum_{i,j}\frac{R_{ij}-R'_{ij}}{R_{ij}}}{w}$	This factor is similar to MAE and indicates the forecast quality.
RC	$RC = \frac{\sum_{u=1}^{T} PR_u}{\sum_{u=1}^{T} S_u}$	A low value of RC indicates the system cannot predict many uncertain missing values. PR_u is the total predicted number of items for test user u S_u is the number of ratings that can be predicted.
iMAE(inverse MAE)	$iMAE = 1 - \frac{MAE}{\max - \min}$	The prediction accuracy normalized by the range of rating scales
MRR(mean reciprocal rank)	$MRR = \frac{1}{\lvert M\rvert}\sum_{m\in M}\left(\sum_{i\in test(m)}\frac{1}{rank(m,i)}\right)$	It is used to measure performance. rank (m, i) indicates the position of service i in the recommender list for mashup m.
Mean squared error (MSE)	$MAE = \frac{\sum_{i,j}(R_{ij}-\widehat{R}_{ij})}{W}$	This factor is similar to MAE.
Memory Complexity	-	The spatial complexity of an algorithm is the amount of space the algorithm occupies relative to the magnitude of the input [31]. It is the quantity of memory required to tackle an example computational problem as a function of the input size.

5.1 Dataset

The data sets used in various articles are described in this section. Most of the studied articles have used Wsdream dataset to test their method. Other articles have also used different datasets, which will be explained in the rest of this section. In this section, 15 datasets are introduced and explained (Table 3).

5.2 Comparison methods

In this section, the methods that used the WSdream dataset are compared. This dataset includes two datasets, Through Put(TP) and Response Time(RT). The first diagram compares Mean absolute error or MAE in the mentioned methods. This comparison is obtained on the TP dataset and when the data sparsity is 20% and 5%. The second diagram calculates the RMSE on the RT dataset when the sparsity is 20% and 5%. The third diagram shows the MAE in the RT dataset in the mentioned methods when the data sparsity is 20 and 5. Diagram 4 shows the RMSE on the TP dataset.

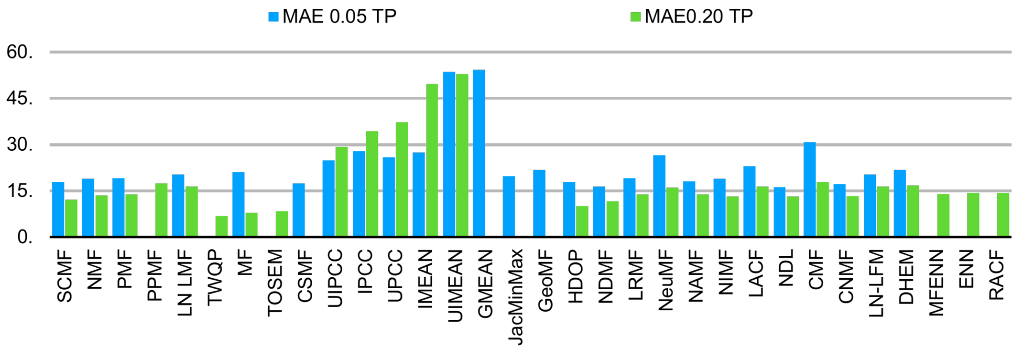

Diagram1. Comparison of mentioned methods on MAE parameter on TP

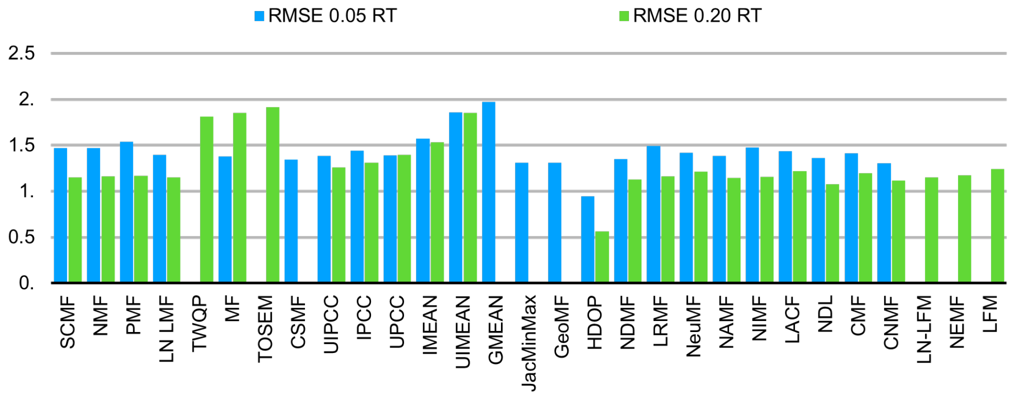

Diagram 2. Comparison RMSE on RT

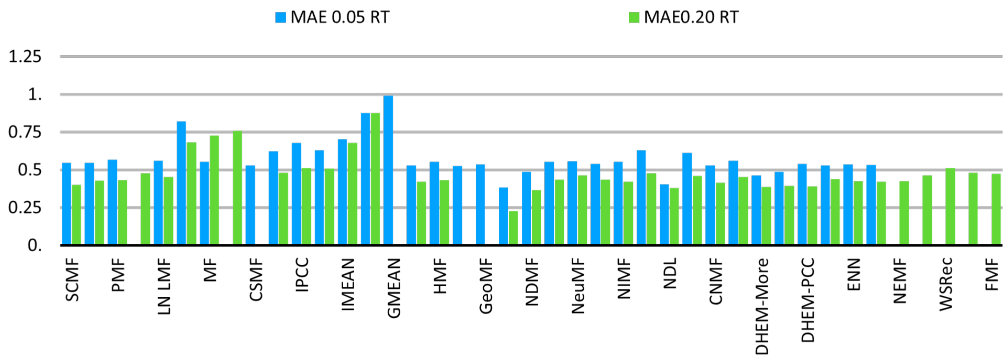

Diagram 3. Comparison MAE RT

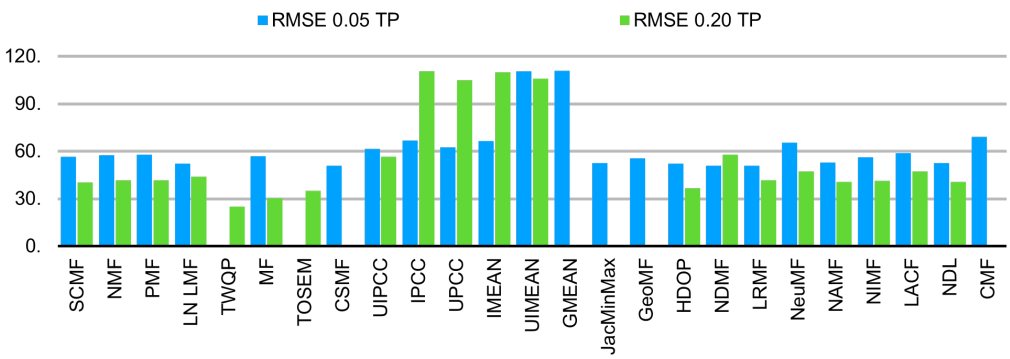

Diagram 4. Comparison RMSE on TP

Table 3: Datasets characteristics

Dataset title	proposed by	Description
WS-DREAM Datasets	Zheng- paper	This repository maintains a set of QoS datasets which collected from real-world Web services. dataset#1 describes real-world QoS measurements, including both response time and throughput values, obtained from 339 users on 5,825 Web services. dataset#2 describes real-world QoS measurements from 142 users on 4,500 Web services over 64 consecutive time slices (at 15-minute interval). Web services features in WS-Dream is: Service ID, WSDL Address, Service Provider, IP Address ,Country, Continent, Latitude, Longitude, Region and City.
BTP dataset	paper 33, Shangguang Wang 2012	This dataset contains QoS data via the Beijing taxi passengers (BTP). It contains information on the number of people called and got off taxies at different time and locations. The raw data contains about one billion records, each of which consists of taxi ID, longitude, latitude, speed, travel direction, carrying passengers or not, sampling time and so on. For a specific taxi, value change of the field carrying passenger or not to Yes from No means a passenger called the taxi at the recorded sampling time. Similarly, changing to No from Yes of the field means a passenger got off the taxi at the recorded sampling time. This dataset extract the data of a week (2012.11.05 to 2012.11.11) . The zone of Beijing divided within Third Ring Road into 3600 grids, each of which is 240m×240m size, as shown in Fig. 1.
MovieLens 100k	collected by the GroupLens Research Project at the University of Minnesota.	MovieLens 100 K Ratings (SML): This data set contains 943 users, 1682 movies, and 100000 ratings on an integer scale 1 (bad) to 5 (excellent). Each user has rated at least 20 movies. This dataset contain simple demographic info for the users (age, gender, occupation, zip). The data was collected through the MovieLens web site(movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set. Its sparsity is 93.7% that shows only 6.3% of the total user-item pairs have been rated(density). There are total 5 classes against which the users have provided the ratings.
Movielens 1M	collected by the GroupLens Research Project at the University of Minnesota paper39	These dataset contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users. Rating dataset contains UserID,MovieID,Rating,Timestamp Ratings are made on a 5-star scale, Each user has at least 20 ratings user dataset contains UserID,Gender,Age,Occupation,Zip-code. Movie dataset contains MovieID,Title,Genres.
Movielens 10M		This data set contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service `MovieLens`. Users were selected at random for inclusion. All users selected had rated at least 20 movies. Unlike previous MovieLens data sets, no demographic information is included. Each user is represented by an id, and no other information is provided.
FilmTrust (FT)	Guo,2013	This dataset created by crawling the Film- Trust website. The dataset retrieved contains 1214 users, 1922 movies, and 28 645 ratings on a floating point scale of 1 (bad) to 10 (excellent) (with a difference of 0.25). Its sparsity is 98:8%. The FilmTrust dataset adequately captures the new user and new item cold-start problems. It has imbalanced data, i.e., one user may have provided one rating and others may have provided hundreds of ratings and the same is true for items of the FilmTrust dataset. they observe that there are total 8 classes against which the users have provided the ratings.
Shanghai Telecom	provided by Shanghai Telecom, paper 28	The Shanghai Telecom dataset contains Internet information more than 7.2 million records of accessing the Interent through 3,233 base stations from 9,481 mobile phones for six months. they contain 6358 users and 3233 edge servers. Every user has a unique ID and multiple edge servers; every edge server owns a detailed location. This dataset could help researchers to evaluate their solution in mobile edge computing topic such as edge server placement, service migration, service recommendation, etc. the Telecom dataset shows 6 parameters such as Month, Data, Start Time, End Time, Base Station Location, Mobile Phone ID. The trajectory of users can be found by the dataset.
WSMonitor	2011	WSMonitor crawls a set of WSDL files from the Internet. WSMonitor deployed on 142 distributed computers located in 22 countries from PlanetLab. 4,532 publicly available real world Web services from 57 countries are monitored. This dataset contain 30,287,611 Web service QoS record.
Jester		4.1 Million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 anonymous users: collected between April 1999 - May 2003. This dataset contains 3 files. First, data from 24,983 users who have rated 36 or more jokes, a matrix with dimensions 24983 X 101. Second, data from 23,500 users who have rated 36 or more jokes, a matrix with dimensions 23500 X 101. Third, data from 24,938 users who have rated between 15 and 35 jokes, a matrix with dimensions 24,938 X 101.

Flixster	This contains the friendship network crawled in December 2010 by Javier Parra	Flixster is a classical recommendation dataset which nearly contains 535013 hard 418 ratings on 19 different types of movies (Drama, Comedy and so on). Flixster is a social movie site allowing users to share movie ratings, discover new movies and meet others with similar movie taste.
Epinions	2011	The Epinions dataset is built form a who-trust-whom online social network of a general consumer review site epinions.com. Members of the site can decide whether to "trust" each other. All the trust relationships interact and form the Web of Trust which is then combined with review ratings to determine which reviews are shown to the user. It contains 75,879 nodes and 50,8837 edges. For each user, dataset have his profile, his ratings and his trust relations. For each rating, this have the product name and its category, the rating score, the time point.
Ciao	2011	The Ciao dataset contains movie rating information of users given to items, and also contain item category information. Users in Ciao can rate products using scores from 1 to 5. 7252 users, 21880 items, 183749 rating, rating density 0.0012
BookCrossing	Cai-Nicolas Ziegler	Dataset collected in a 4-week crawl (August / September 2004) from the `Book-Crossing` community with kind permission from Ron Hornbaker, CTO of Humankind Systems. It contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit / implicit) about 271,379 books. BX-Users, Contains the users. Note that user IDs have been anonymized and map to integers. Demographic data is provided (*'Location'*, *'Age'*) if available. Otherwise, these fields contain *NULL*-values. BX-Books, the books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (*'Book-Title'*, *'Book-Author'*, *'Year-Of-Publication'*, *'Publisher'*), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (*'Image-URL-S'*, *'Image-URL-M'*, *'Image-URL-L'*), i.e., small, medium, large. These URLs point to the Amazon web site.BX-Book-Ratings, contains the book rating information. Ratings (*'Book-Rating'*) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.
Lastfm	`http://www.last.fm/api/tos`	This dataset collection of song-level tags and precomputed song-level similarity. This dataset contains <user, artist, plays> tuples (for $\sim$360,000 users) collected from Last.fm API There are 122877 tracks and 100 tags in the dataset.
Pinterest	Universidad de Guanajuato - Campus Irapuato-Salamanca	The Pinterest dataset contains more than 1 million images associated to Pinterest users' who have "pinned" them. This dataset, contains 70,200 pins belonging to 117 users, that were randomly collected by directly crawling the Pinterest website .

6 Conclusion

Recommender systems provide users with items depending on their requirements. Since several cloud services are present, there is a need to propose the correct one to users according to their preferences. Service quality values should be predicted correctly due to their critical role in the quality of cloud recommender systems. Because the history of users' choices and previous selections by them are not available, it is challenging to recommend services to them. Numerous approaches proposed to correctly forecast service quality values aim at improving the recommender system quality. Collaborative filtering, matrix factorization, and clustering are some of these techniques. Initially, this review paper raises a general problem, and then a statement follows that is required for the research. Further discussions are the types of recommender systems, as well as their challenges and problems. This review evaluates different methods, and platforms, solutions, with their pros and cons. Several graphs compare the values obtained using various approaches of the papers. All datasets used in the papers are discussed and introduced in this SLR. Furthermore, a detailed compilation is offered of the whole metrics and formulas employed in the papers.

References

[1] C.C. Aggarwal and C.C. Aggarwal, *Social and trust-centric recommender systems*, Recomm. Syst.: The Textbook (2016), pp. 345–384.

[2] M. Aghaei, P. Asghari, S. Adabi, and H. Haj Seyyed Javadi, *Using recommender clustering to improve quality of services with sustainable virtual machines in cloud computing*, Cluster Comput. **26** (2023), no. 2, 1479–1493.

[3] Z. Ali, S. Khusro, and I. Ullah, *A hybrid book recommender system based on table of contents (toc) and association rule mining*, Proc. 10th Int. Conf. Inf. Syst., 2016, pp. 68–74.

[4] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, *Data mining methods for recommender systems*, Recommender systems handbook, Springer, 2010, pp. 39–71.

[5] B. Barzegar, H. Motameni, and A. Movaghar, *Eatsdcd: A green energy-aware scheduling algorithm for parallel task-based application using clustering, duplication and dvfs technique in cloud datacenters*, J. Intell. Fuzzy Syst. **36** (2019), no. 6, 5135–5152.

[6] J. Basiri, A. Shakery, B. Moshiri, and M. Zi Hayat, *Alleviating the cold-start problem of recommender systems using a new hybrid approach*, 5th Int. Symp.Telecommun., IEEE, 2010, pp. 962–967.

[7] J. Basiri, A. Shakery, B. Moshiri, and M. Zi Hayat, *Addressing the new user cold-start problem in recommender systems using ordered weighted averaging operator*, Int. J. Inf. Commun.Technol. Res. **2** (2010), no. 4, 79–87.

[8] J. Beel, C. Breitinger, S. Langer, A. Lommatzsch, and B. Gipp, *Towards reproducibility in recommender-systems research*, User Model. User-adapted Interact. **26** (2016), 69–101.

[9] H. Bouazza, B. Said, and F. Zohra Laallam, *A hybrid IoT services recommender system using social IoT*, J. King Saud Univ.-Comput. Inf. Sci. **34** (2022), no. 8, 5633–5645.

[10] L. Cao, *Non-iid recommender systems: A review and framework of recommendation paradigm shifting*, Engineering **2** (2016), no. 2, 212–224.

[11] W. Carrer-Neto, M. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez, *Social knowledge-based recommender system. application to the movies domain*, Expert Syst. Appl. **39** (2012), no. 12, 10990–11000.

[12] Y. Ho Cho, J. Kim, and S. Hie Kim, *A personalized recommender system based on web usage mining and decision tree induction*, Expert Syst. Appl. **23** (2002), no. 3, 329–342.

[13] B. Deebak and F. Al-Turjman, *A novel community-based trust aware recommender systems for big data cloud service networks*, Sustain. Cities Soc. **61** (2020), 102274.

[14] M. D. Ekstrand, J.T. Riedl, J.A. Konstan, *Collaborative filtering recommender systems*, Found. Trends Human–Comput. Interact. **4** (2011), no. 2, 81–173.

[15] M. Etemadi, S. Bazzaz Abkenar, A. Ahmadzadeh, M. Haghi Kashani, P. Asghari, M. Akbari, and E. Mahdipour, *A systematic review of healthcare recommender systems: Open issues, challenges, and techniques*, Expert Syst. Appl. **213** (2023), 118823.

[16] S. Fatehi, *Task scheduling optimization based on heuristic algorithm for heterogeneous cloud computing platforms*, Int. J. Nonlinear Anal. Appl. **13** (2022), no. 2, 2743–2750.

[17] S. Fatehi, H. Motameni, B. Barzegar, and M. Golsorkhtabaramiri, *Energy aware multi objective algorithm for task scheduling on dvfs-enabled cloud datacenters using fuzzy nsga-ii*, Int. J. Nonlinear Anal. Appl.**12** (2021), no. 2, 2303–2331.

[18] D. Gavalas, V. Kasapakis, C. Konstantopoulos, K. Mastakas, and G. Pantziou, *A survey on mobile tourism recommender systems*, Third Int. Conf. Commun. Inf. Technol., IEEE, 2013, pp. 131–135.

[19] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, *Mobile recommender systems in tourism*, J. network Comput. Appl. **39** (2014), 319–333.

[20] M. Ghobakhloo and M. Ghobakhloo, *Design of a personalized recommender system using sentiment analysis in social media (case study: Banking system)*, Soc. Network Anal. Min. **12** (2022), no. 1, 84.

[21] L. Guo, B. Jin, C. Yao, H. Yang, D. Huang, F. Wang, *Which doctor to trust: A recommender system for identifying the right doctors*, J. Med. Internet Res. **18** (2016), no. 7, e6015.

[22] V.L. Hallappanavar, C.M. Bulla, and MN. Birje, *Ann based estimation of reputation of newcomer web services in fog computing*, Int. Conf. Comput. Commun. Inf., IEEE, 2021, pp. 1–7.

[23] Y. Himeur, A. Alsalemi, A. Al-Kababji, F. Bensaali, A. Amira, C. Sardianos, G. Dimitrakopoulos, and I. Varlamis, *A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects*, Inf. Fusion **72** (2021), 1–21.

[24] P. Kosmides, C. Remoundou, K. Demestichas, I. Loumiotis, E. Adamopoulou, and M. Theologou, *A location recommender system for location-based social networks*, Int. Conf. Math. Comput. Sci. Ind., IEEE, 2014, pp. 277–

280.

[25] G. Liang, C. Sun, J. Zhou, F. Luo, J. Wen, and X. Li, *A general matrix factorization framework for recommender systems in multi-access edge computing network*, Mobile Networks Appl. **27** (2022), no. 4, 1629–1641.

[26] W. Liang, S. Xie, J. Cai, J. Xu, Y. Hu, Y. Xu, and M. Qiu, *Deep neural network security collaborative filtering scheme for service recommendation in intelligent cyber–physical systems*, IEEE Internet Things J. **9** (2021), no. 22, 22123–22132.

[27] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, *Recommender system application developments: A survey*, Decision Support Syst. **74** (2015), 12–32.

[28] Z. Ma, M. H. Nejat, H. Vahdat-Nejad, B. Barzegar, and S. Fatehi, *An efficient hybrid ranking method for cloud computing services based on user requirements*, IEEE Access **10** (2022), 72988–73004.

[29] J. Masoudi, B. Barzegar, and H. Motameni, *Energy-aware virtual machine allocation in dvfs-enabled cloud data centers*, IEEE Access **10** (2021), 3617–3630.

[30] S.E. Middleton, N.R. Shadbolt, and D.C. De Roure, *Ontological user profiling in recommender systems*, ACM Trans. Inf. Syst. **22** (2004), no. 1, 54–88.

[31] S.H. Min and I. Han, *Recommender systems using support vector machines*, Int. Conf. Web Engin., Springer, 2005, pp. 387–393.

[32] M.H. Nejat, H. Motameni, H. Vahdat-Nejad, and B. Barzegar, *Efficient cloud service ranking based on uncertain user requirements*, Cluster Comput. (2022), 1–18.

[33] Badieh. Nikzad, B. Barzegar, and H. Motameni, *Sla-aware and energy-efficient virtual machine placement and consolidation in heterogeneous dvfs enabled cloud datacenter*, IEEE Access **10** (2022), 81787–81804.

[34] Z. Peng, B. Barzegar, M. Yarahmadi, H. Motameni, and P. Pirouzmand, *Energy-aware scheduling of workflow using a heuristic method on green cloud*, Sci. Program. **2020** (2020), 1–14.

[35] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, *Ears: Emotion-aware recommender system based on hybrid information fusion*, Inf. Fusion **46** (2019), 141–146.

[36] T. Shao, X. Yang, F. Wang, C. Yan, and A. Kr Luhach, *Trusted service evaluation for mobile edge users: Challenges and reviews*, Complexity **2021** (2021), 1–10.

[37] B. Shapira, L. Rokach, and F. Ricci, *Recommender systems: Techniques, applications, and challenges*, Recommender systems handbook, Springer, 2021, pp. 1–35.

[38] C. Sharma and P. Bedi, *Ccfrs–community based collaborative filtering recommender system*, J. Intell. Fuzzy Syst. **32** (2017), no. 4, 2987–2995.

[39] S.K. Shinde and U. Kulkarni, *Hybrid personalized recommender system using centering-bunching based clustering algorithm*, Expert Syst. Appl. **39** (2012), no. 1, 1381–1387.

[40] J. Son and S. Bum Kim, *Academic paper recommender system using multilevel simultaneous citation networks*, Decision Support Syst. **105** (2018), 24–33.

[41] J. Sun, Z. Wang, X. Luo, P. Shi, W. Wang, L. Wang, J.H. Wang, and W. Zhao, *A parallel recommender system using a collaborative filtering algorithm with correntropy for social networks*, IEEE Trans. Network Sci. Engin. **7** (2018), no. 1, 91–103.

[42] Q. Zhang, J. Lu, and Y. Jin, *Artificial intelligence in recommender systems*, Complex Intell. Syst. **7** (2021), 439–457.

[43] Y. Zhang, *Grorec: A group-centric intelligent recommender system integrating social, mobile and big data technologies*, IEEE Trans. Serv. Comput. **9** (2016), no. 5, 786–795.

[44] D. Zhong, G. Yang, J. Fan, B. Tian, and Y. Zhang, *A service recommendation system based on rough multidimensional matrix in cloud-based environment*, Comput. Standards Interfaces **82** (2022), 103632.